

代码的肉身：从 Monorepo 到真正产品的蜕变

独立开发者日记 (Day 5-10) —— 测试、品牌
与小鱼干经济学

目标：让冰冷的代码库看起来、用起来像个真正的产品。


```
git log --oneline --since="Day 5"
d41d8cd refactor: optimize virtual list component
a0b2c3d fix: resolve styling issues in header
e5f6g7h feat: implement user profile schema
i8j9k0l style: update color palette variables
m1n2o3p test: add unit tests for data fetching
q4r5s6t chore: update dependencies and lock file
u7v8w9x refactor: simplify state management
y0z1a2b fix: address minor bug in form validation
c3d4e5f feat: integrate initial analytics tracking
g6h7i8j test: setup basic e2e testing environment
```

4 天 | 147 个 Commit

过程不算光鲜，但有了底气。一个人狂奔，必须先建好护城河。

阶段一：骨架 (Infrastructure) - 铺满 Vitest

API 路由 (API Routes)

集成测试 (Integration Test) —— Mock Token, 验证扣费与 SSE 格式。

领域计算 (Domain Logic)

八字推算、塔罗抽牌、六壬起卦。

15条 Streaming 路由全压在这! 如果有 Race Condition, 用户要么白嫖, 要么被双重扣费。

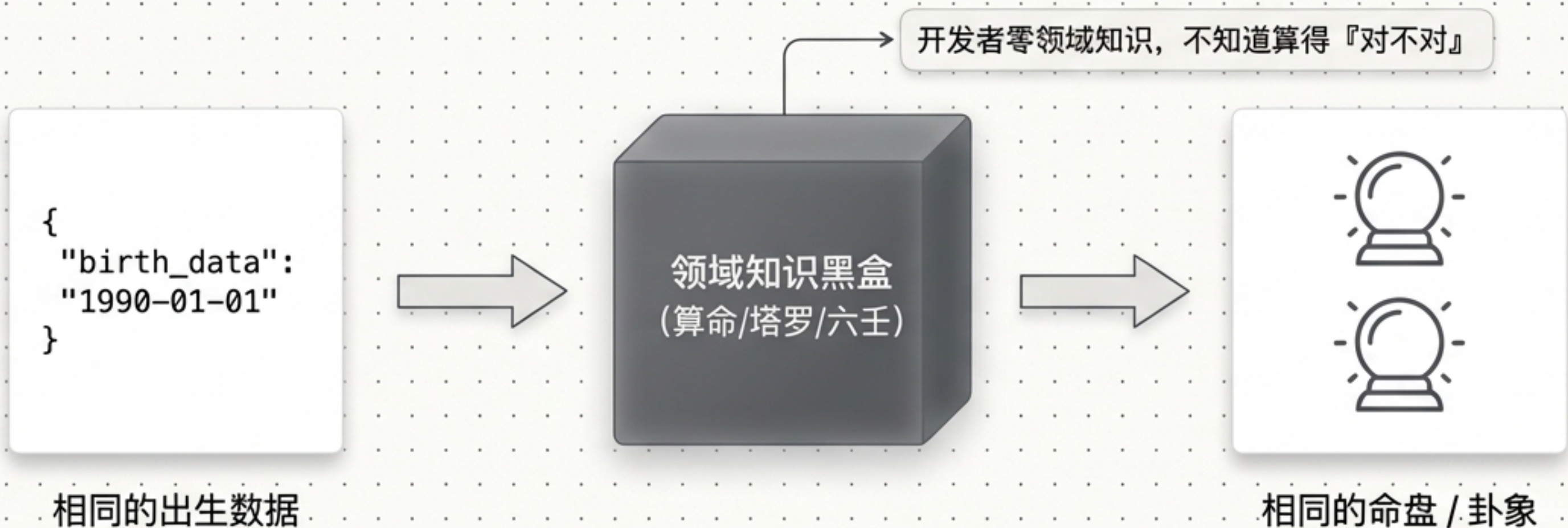
共享包 (Shared Packages)

核心基石: API 流处理、Supabase 客户端、积分管理。

`createAIStreamResponse()` —— 单个工具覆盖 36 个测试用例。



验证确定性 ≠ 验证正确性



同一次洗牌分布符合预期, 同一个卦象含义不变。测确定性, 不需要懂算命。

单人开发的第二双眼睛



每个 PR 走完整流程。一个人做开发，最怕改一处崩另一处。
有了 CI 盯着，后续迭代再快也不怕。

骨架之上，长出皮相与灵魂



给一只算命的猫付小鱼干，
这个设定顺理成章。

Property ×

Brand Attributes

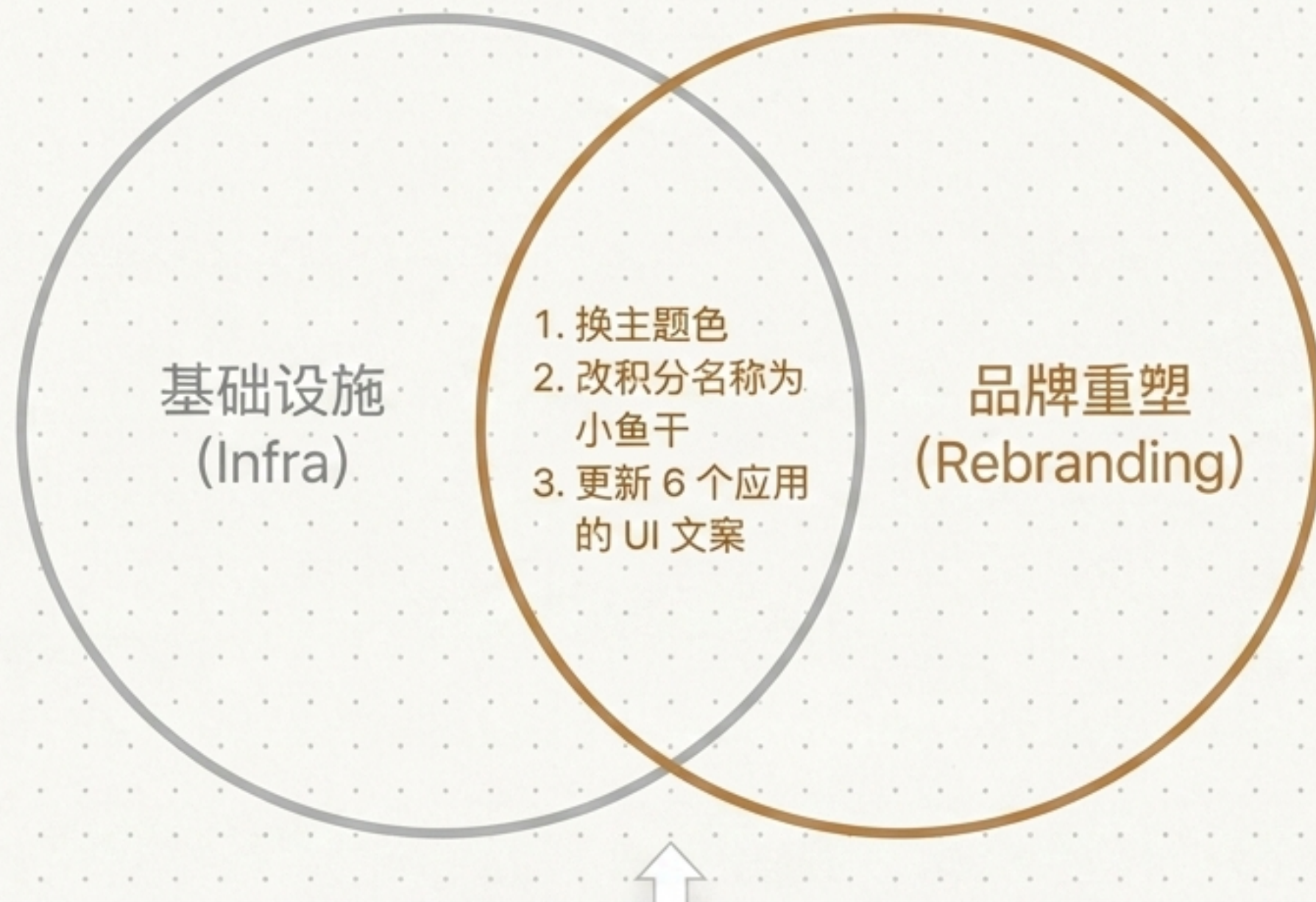
主题色 (Theme):
暖棕与金色  

结算单位 (Currency):
~~积分~~ →  小鱼干

品牌基调 (Tone):
亲切、好玩

维度 (Dimension)	传统算命应用 (Traditional AI)	盼盼猫 (PanPanMao Theme)
用户心理 (Psychology)	庄重、吓人、可能有防备心	亲切、好玩、有趣 (降低试用门槛)
社交动机 (Social Sharing)	纠结“算得准不准”	觉得“这个猫咪设定挺有趣” (促进裂变传播)
迭代宽容度 (Tolerance)	出现 Bug 会被认为“骗子”	猫咪算命，容错率天然更高

核心洞察：品牌从来不仅是视觉，品牌本身就是最高级的产品决策。



**这一圈改下来，测试套件帮我抓出了
3 个差点上线的回归 Bug。**

前面铺测试的投入，
到这里已经全额回本。

美元 (\$USD) – 用户心理防御极高

积分 (Credits) –
抽象层，削弱了具体
金额感

小鱼干 (Dried Fish) –
游戏化体验，消费阻力降至最低



每次解读只花几条小鱼干。花小鱼干的时候，脑子里想的根本不是『花钱』这件事。



思考套餐命名花的时间，比写整个 Stripe 支付流程的代码还要长。

单人创业者的杠杆

这六天的本质不是写了 147 个 Commit，而是通过系统设计，
将单人开发者的时间成本、用户的心理防线、消费的痛觉，全部“降维”了。

工程杠杆

全面的
Vitest & CI

极速迭代且不崩溃的底气

增长杠杆

盼盼猫
品牌包装

极低的用户分享与信任摩擦力

营收杠杆

小鱼干代币
+
价格锚点

极低的消费痛觉



不过这版后来被
彻底推翻了。

『温馨』的调性，削弱了用户对一个算命平台的庄重感。

下一期（第 12-19 天）：推翻重来。走向暗黑奢华风，以及变现的深水区。